UNMASKING ENCRYPTION: SIDE-CHANNEL ANALYSIS ON AES USING CPA

Name: AMAN SHUKLA

Course:ECMM451

SID: 730095963

BACKGROUND & CONTENT

- **AES Encryption**: Symmetric encryption standard (NIST, 2001).
- **Power analysis Side-Channel Attack**: Exploits physical leakage (power, EM) to recover keys.
- **Historical Research**: Loic Masure's ASCAD project for ANSSI, has demonstrated the effectiveness of these attacks in compromising secure systems.
- **Signal-to-Noise Ratio (SNR)**: Compares the level of signal leakage to background noise, essential for effective analysis in Correlation Power Analysis (CPA).
- **Correlation Power Analysis (CPA)**: Statistically correlates power consumption patterns with hypothesized key values to retrieve cryptographic keys.
- **Masking** : Obscures sensitive data by adding random values to prevent side-channel leakage.
- **Intermediate value** : Masked or unmasked variables in the AES process analyzed for information leakage during power analysis attacks
- **Permutation Indices**: Used to reorder data to further obscure patterns and enhance security against attacks.
- **S-Box**: A non-linear transformation in AES that substitutes each plaintext byte with a corresponding byte, enhancing encryption confusion.

- **Pearson's Correlation coefficient:**

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sqrt{E[(X-\mu_X)^2]E[(Y-\mu_Y)^2]}}$$

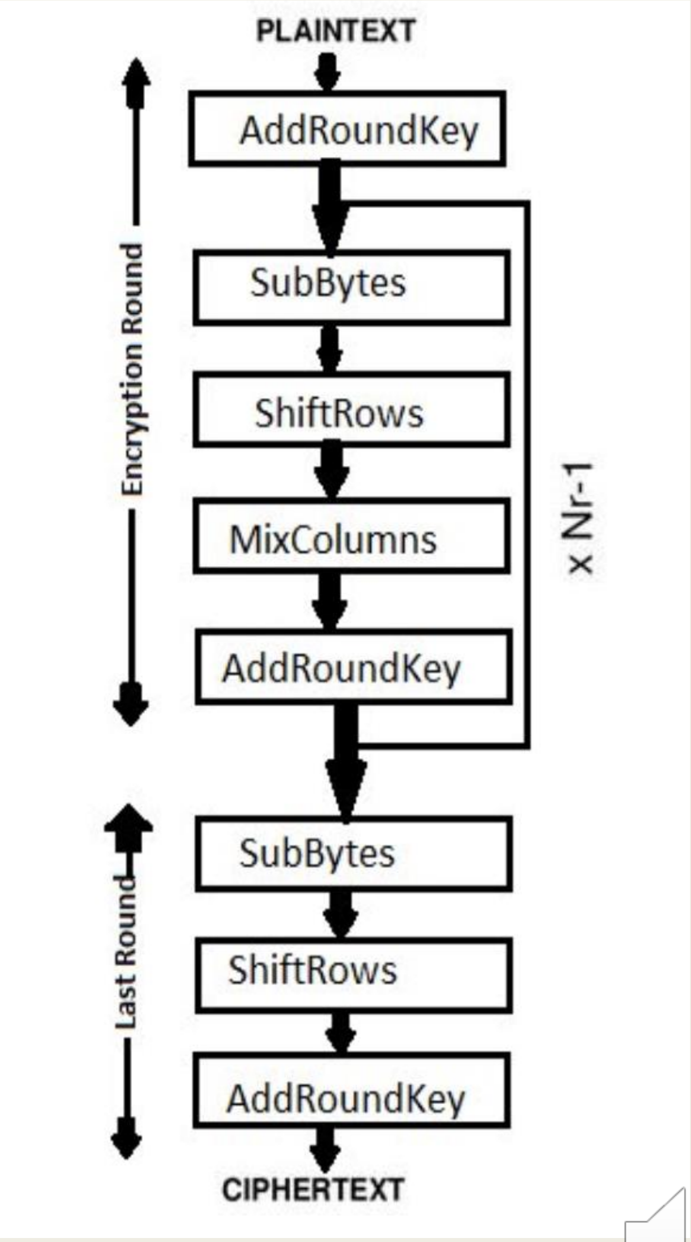


Figure: Flow chart of AES

Hypothesis

SNR analysis, when integrated with Correlation Power Analysis (CPA), significantly enhances the accuracy of key recovery in masked AES implementations by effectively identifying and exploiting Points of Interest (PoIs) within power traces.

Aims & Objectives

Identify

Identify PoIs: Using SNR analysis to find points in the power traces that correlate with the secret key.

Perform

Perform CPA: Using identified PoIs to execute first and second-order CPA attacks.

Evaluate

Evaluate Effectiveness: Assessing the success rate and guessing entropy to measure attack effectiveness.

Dataset overview

Dataset Source

- This database contained the power consumption of a STM32 Cortex M4 microcrontroller (STM32F303RCT7) for AES encryption.
- Taken from the website of ANSSI France, created by Loïc Masure et al., known as ASCADv2_extracted. Link- https://www.data.gouv.fr/en/datasets/ascadv2/

ASCADv2_extracted contents:

- Contains 15,000 power traces, a subset of the larger 800,000 trace dataset, sufficient for analyzing one AES cycle. Dataset includes power traces, meta data and labels.
- **Metadata:** Includes plaintexts, keys, and masks used during AES encryption, essential for constructing CPA hypotheses.
- **Labels:** Contains alpha masks, beta masks, masked S-box values, and permutation indices to understand masking effects.

EXPERIMENT DESIGN

Trace Mean Calculation

- **Purpose:** To understand the general power consumption pattern.
- **Trace Mean Graph**: The peaks potentially indicating AES operations like SubBytes or MixColumns, helping identify Points of Interest (PoIs) for further Correlation Power Analysis (CPA).

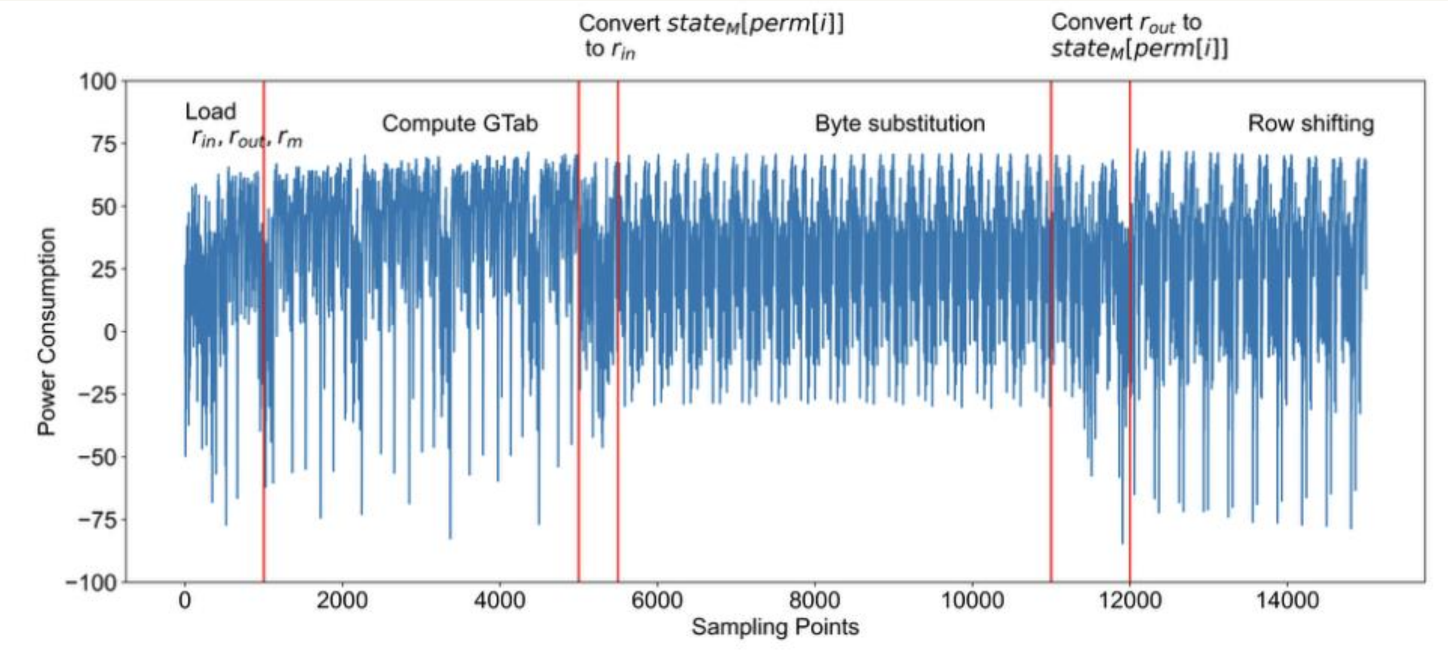


Figure: Trace Mean Graph

SNR Analysis

```
SNR_inds_ = RunningSNR(n_classes=16)     # Permutation indices
SNR_r_in_ = RunningSNR(n_classes=256)    # Boolean mask r_in
SNR_r_out_ = RunningSNR(n_classes=256)   # Boolean mask r_out
SNR_r_m_ = RunningSNR(n_classes=255)     # Multiplicative mask r_m
SNR_c_ = RunningSNR(n_classes=256)       # Affine mask protected intermediate state value
SNR_c1_ = RunningSNR(n_classes=256)      # Multiplicative mask protected intermediate state value
SNR_c2_ = RunningSNR(n_classes=256)      # Unprotected intermediate state value
trace_mean_ = RunningMean()              # Power trace mean
```

Code: Initialization of SNR

SNR (Signal-to-Noise Ratio) analysis identifies Points of Interest (PoIs) by quantifying signal distinguishability against background noise in power traces

Analyzing Permutation indices, r_in, r_out, r_m, c, c1, c2, SNR highlights where power variations correlate with cryptographic operations, guiding targeted CPA.

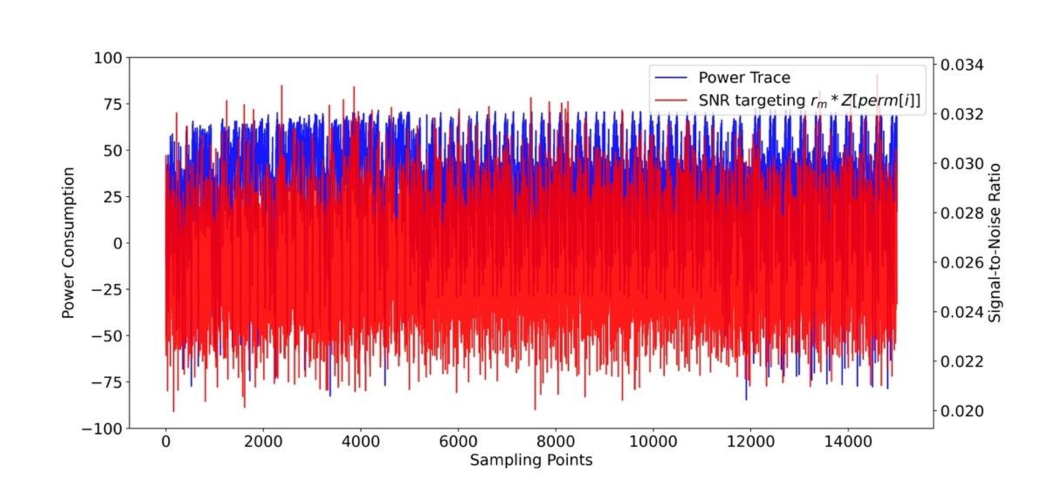


Figure: Plotting SNR for c1(r_m*$Z[perm[i]]$)

```
max SNR_c: 0.07650343885714773
(array([5552, 5553]),)
max SNR for r_out:  0.5727718948568803
(array([  338,   339,   341,   342,   343,   345, 11418, 11419, 11421,
       11425]),)
max SNR for rm:  0.701092019046882
(array([813, 820, 861, 867, 868]),)
```

Figure: Output for max SNR for c, r_out, rm

Corelation Power Analysis (First Order)

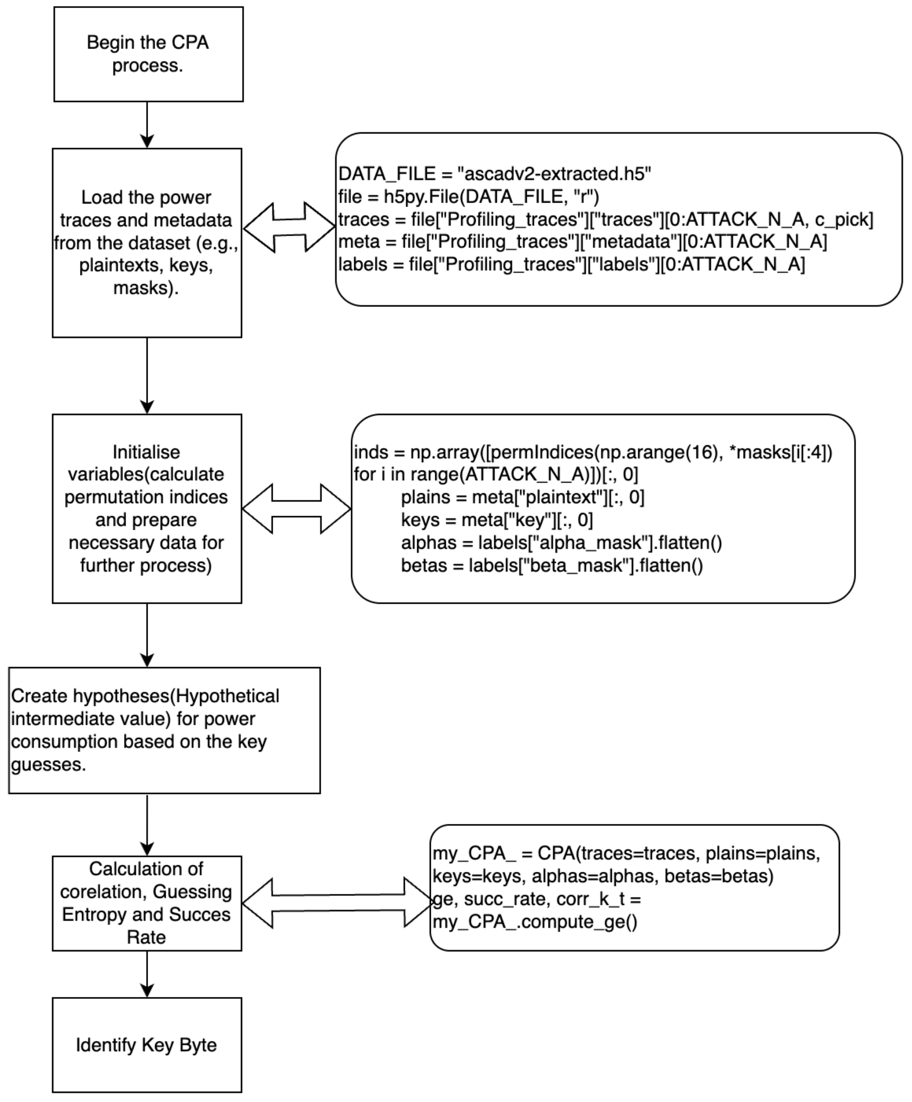


Figure: Flow chart of first order CPA

- **Purpose**: First-Order Correlation Power Analysis is employed to recover key bytes by exploiting the correlation between power consumption and data-dependent operations during AES encryption, specifically targeting the S-box output.
- **How it Works**: It calculates the correlation between actual power traces and hypothesized power consumption for different key guesses, identifying the correct key byte with the highest correlation.

Output: Guessing entropy and success rate.

Corelation Power Analysis(Second Order)

- **Purpose**: This enhances key recovery by targeting more complex masking schemes, particularly focusing on intermediate values that are protected by multiple layers of masking.
- **How it Works**: It analyzes the correlation between combined leakage traces (c and r_out) and hypothesized values derived from these intermediate states, providing a more robust attack against masked implementations.
- **Importance:** This reveals how second-order side-channel attacks can effectively bypass advanced masking techniques, underscoring vulnerabilities in cryptographic implementations.

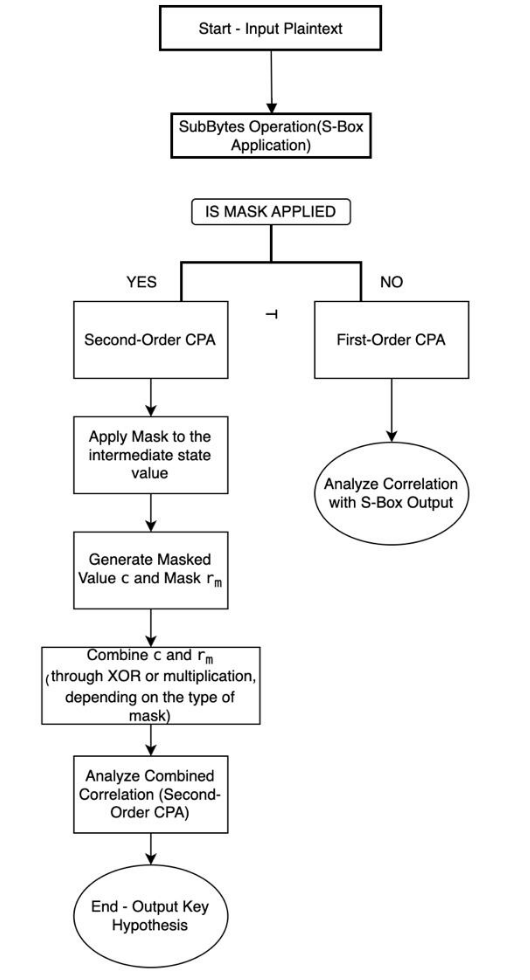
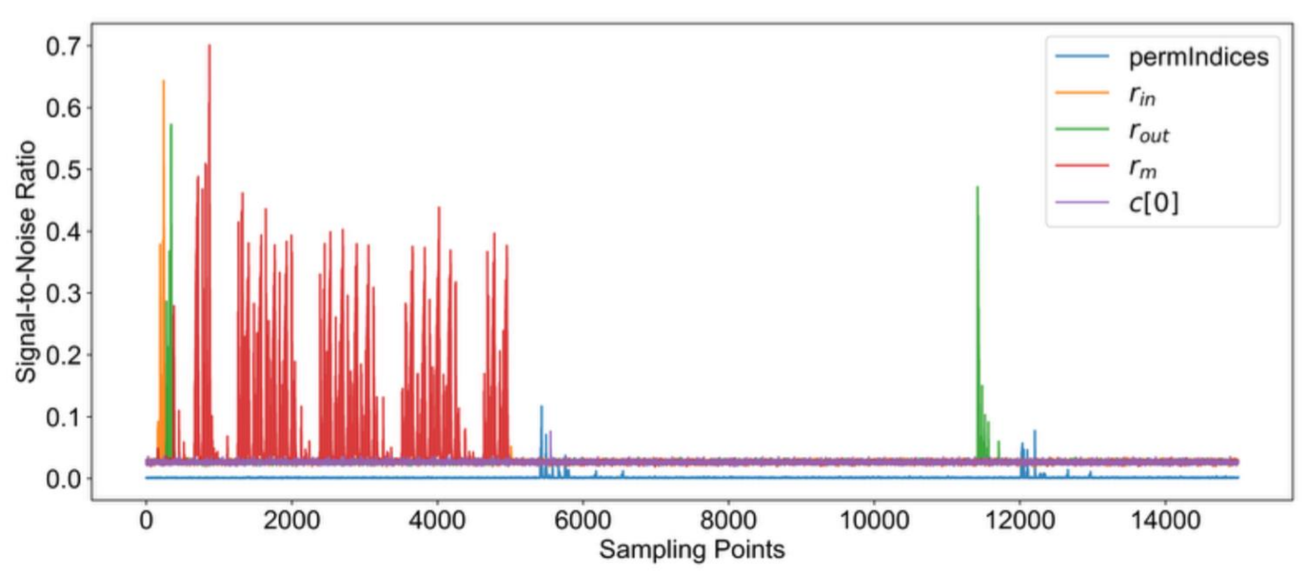


Figure: Flow chart CPA (first vs second order

RESULTS



SIGNAL TO NOISE RATIO ANALYSIS

- Provided a comprehensive view of SNR values across all intermediate values, aiding in selecting the best PoIs for CPA.

CORELATION POWER ANALYSIS (FIRST ORDER)

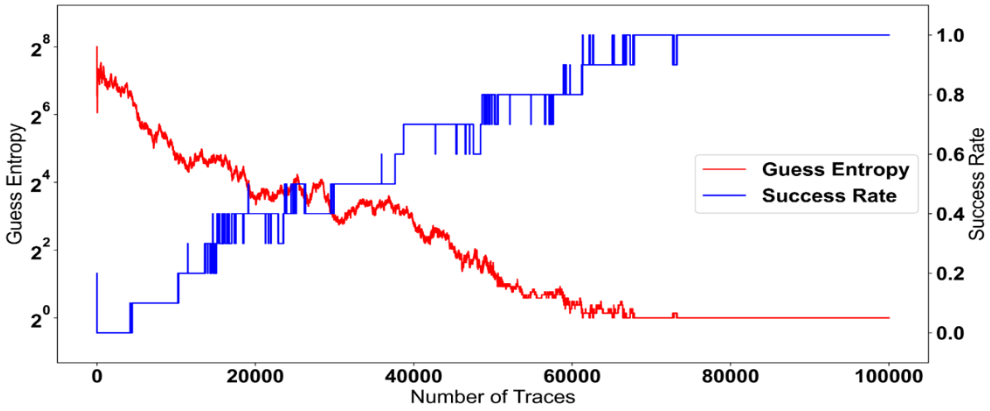


Figure: Guess entropy & Success rate vs No. of traces

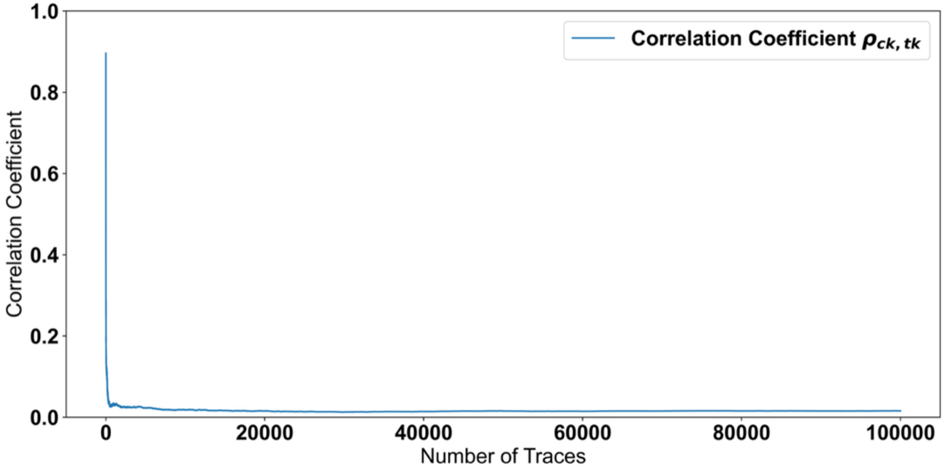


Figure: Correlation Coefficient vs No. of traces

```
Processing trace 99999          Guess entropy [256.  256.   96.1 ...   1.    1.    1. ]
Success rate [0. 0. 0. ... 1. 1. 1.]
Max Correlation coefficient [0.01563903 0.01564591 0.01564321 0.01564521 0.01564237 0.01564409
 0.01563974 0.01563768 0.01564002 0.01564075]
```

Figure: Output showing Guess entropy, Success rate, Max Correlation

- Successful key recovery with first-order analysis, showing the effectiveness of CPA on the identified PoIs.

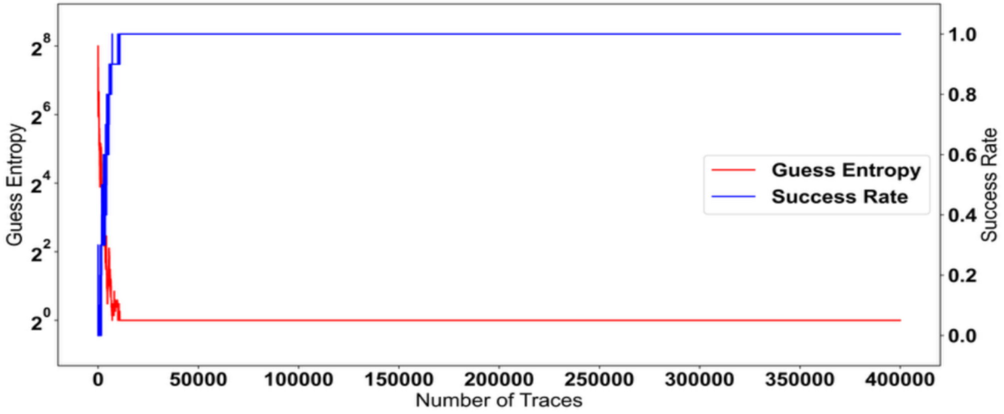


Figure: Guess entropy & Success rate vs No. of traces

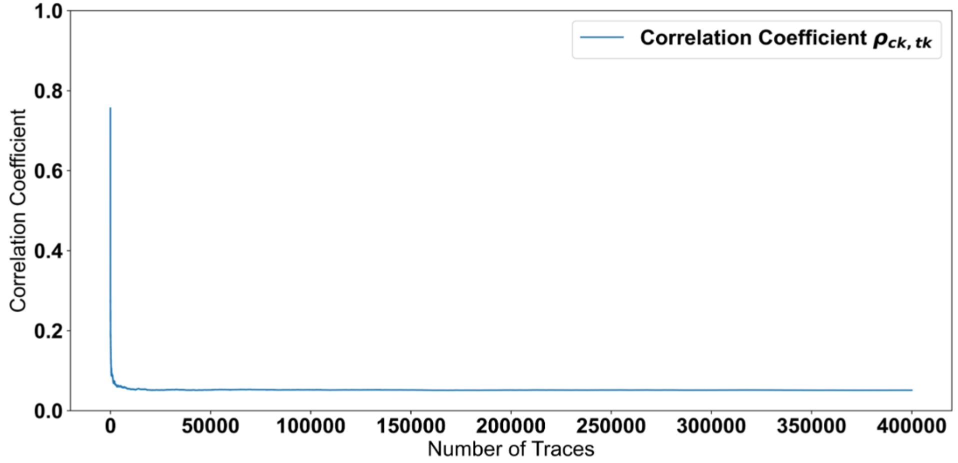


Figure: Correlation Coefficient vs No. of. traces

```
CPA running time: 1744.0783879756927 seconds
[256.   142.3 105.3 ...    1.     1.     1. ]
[0.   0.2 0.3 ... 1.   1.   1. ]
max corelation: [0.05099598 0.05099661 0.05099769 0.05099763 0.05099859 0.05099803
 0.05099796 0.05099893 0.05099888 0.0509992 ]
```

Figure: Output showing running time, Guess entropy, Success Ratio, Max corelation

CORELATION POWER ANALYSIS (SECOND ORDER)

- Enhanced key recovery with second-order analysis, demonstrating improved attack efficiency on more complex masking schemes. Corelation coefficient is ‘0.05100’.

Trace_estimation

```
import numpy as np

# z_a 99.99% = 3.719
def trace_estimate(corr_k_t, z_a=3.719):
    temp = np.log((1 + corr_k_t) / (1 - corr_k_t))
    temp = temp ** 2
    n = 3 + 8 * (z_a ** 2 / temp)
    return n

corr_k_t = 0.05100

print(trace_estimate(corr_k_t=corr_k_t))
```

```
10619.66425791805
```

Trace Estimate

- Provided the number of traces required for reliable correlation coefficient.
- Insight into the efficiency and effectiveness of the method. Estimate= 10619 traces required.

Conclusion:

- The project successfully identified key PoIs using SNR analysis and performed effective CPA to recover the encryption key.
- Demonstrated an effective side-channel attack methodology for AES key recovery.
- Second-order CPA shows better performance in handling masked values.
- SNR analysis is crucial for identifying PoIs.
- The results highlight the vulnerability of masked AES implementations to CPA, even with advanced masking schemes.

Future Work & References

Next Steps:

- The methodology can be further improved for higher accuracy.
- Explore third-order CPA for even more complex masking schemes.
- Improve SNR analysis techniques

Presentation link : Presentation CPA.mp4. Presentation CPA.mp4

References:

- Masure, L. (n.d.). ASCADv2 Code Analysis. GitHub. Link
- National Institute of Standards and Technology (NIST). (2001). Advanced Encryption Standard (AES). FIPS PUB 197.Mangard, S., Oswald, E., & Popp, T. (2007).
- “Power Analysis Attacks: Revealing the Secrets of Smart Cards” by Stefan Mangard, Elisabeth Oswald and Thomas Popp Springer, 2007 ISBN: 978-0-387-30857-9 Arnaud Tisserand CNRS, IRISA Laboratory, Lannion, France
Include links to the tools and scripts developed for this project.

THANK YOU